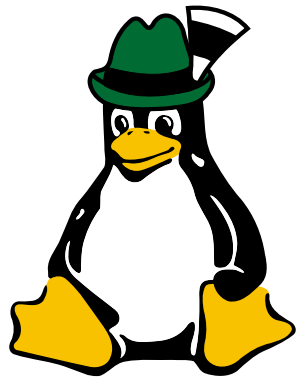
Ein Blick auf SSH

Florian Preinstorfer

https://nblock.org

Grazer Linuxtage
29.04.2017



SSH @ Grazer Linuxtage

- 2013: “Was, SSH kann auch das?”
- 2015: “The Love SSHag”
- 2017: “SSH-From Zero to Hero”

Suche nach “SSH” . . .

- Server: OpenSSH, Dropbear, Apache Mina, CopSSH, . . .

Suche nach “SSH” . . .

- Server: OpenSSH, Dropbear, Apache Mina, CopSSH, . . .
- Clients: OpenSSH, Dropbear, Putty, ConnectBot, . . .

Suche nach “SSH” . . .

- Server: OpenSSH, Dropbear, Apache Mina, CopSSH, . . .
- Clients: OpenSSH, Dropbear, Putty, ConnectBot, . . .
- Konfigurationsanleitungen, Tipps & Tricks

SSHv1 SSHv2

SSHv1

SSHv2

SSH

Secure Shell (SSH) is a protocol for secure remote login and other secure network services over an insecure network.

— RFC 4251

Ziele

- Ein sicheres Netzwerkprotokoll

Ziele

- Ein sicheres Netzwerkprotokoll
- Ablöse von telnet, rlogin, ...

Ziele

- Ein sicheres Netzwerkprotokoll
- Ablöse von telnet, rlogin, ...
- Ein möglichst schlankes Protokoll

Ziele

- Ein sicheres Netzwerkprotokoll
- Ablöse von telnet, rlogin, ...
- Ein möglichst schlankes Protokoll
- Modular & erweiterbar

RFCs

- Kern: RFC4250–RFC4254 (5)

RFCs

- Kern: RFC4250–RFC4254 (5)
- Erweiterungen: RFC4255, RFC4256, RFC4335, RFC4344, RFC4345, RFC4419, RFC4462, RFC4716, RFC5656, RFC6594, RFC6668, RFC7479 (12)

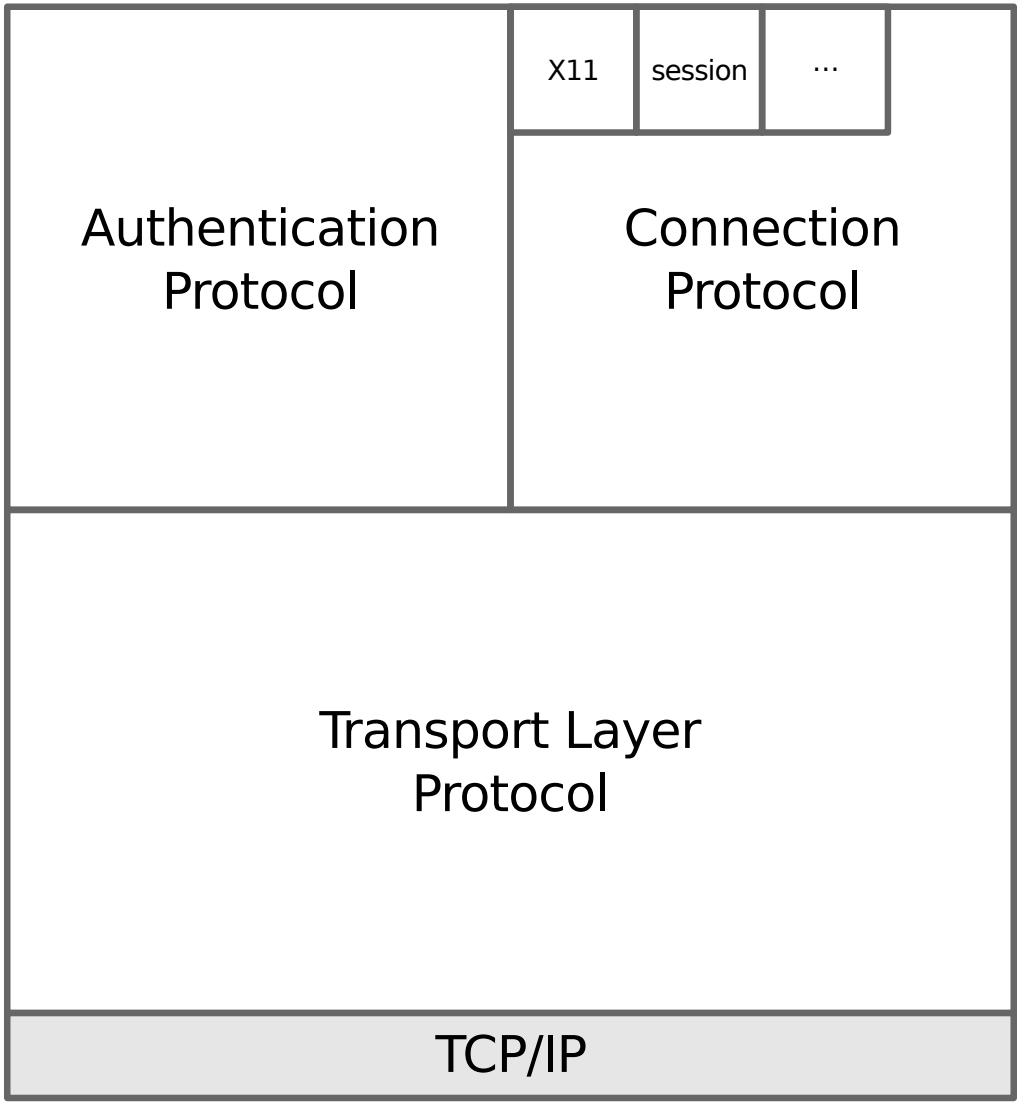
RFCs

- Kern: RFC4250–RFC4254 (5)
- Erweiterungen: RFC4255, RFC4256, RFC4335, RFC4344, RFC4345, RFC4419, RFC4462, RFC4716, RFC5656, RFC6594, RFC6668, RFC7479 (12)
- Erweiterungen (Entwurf)

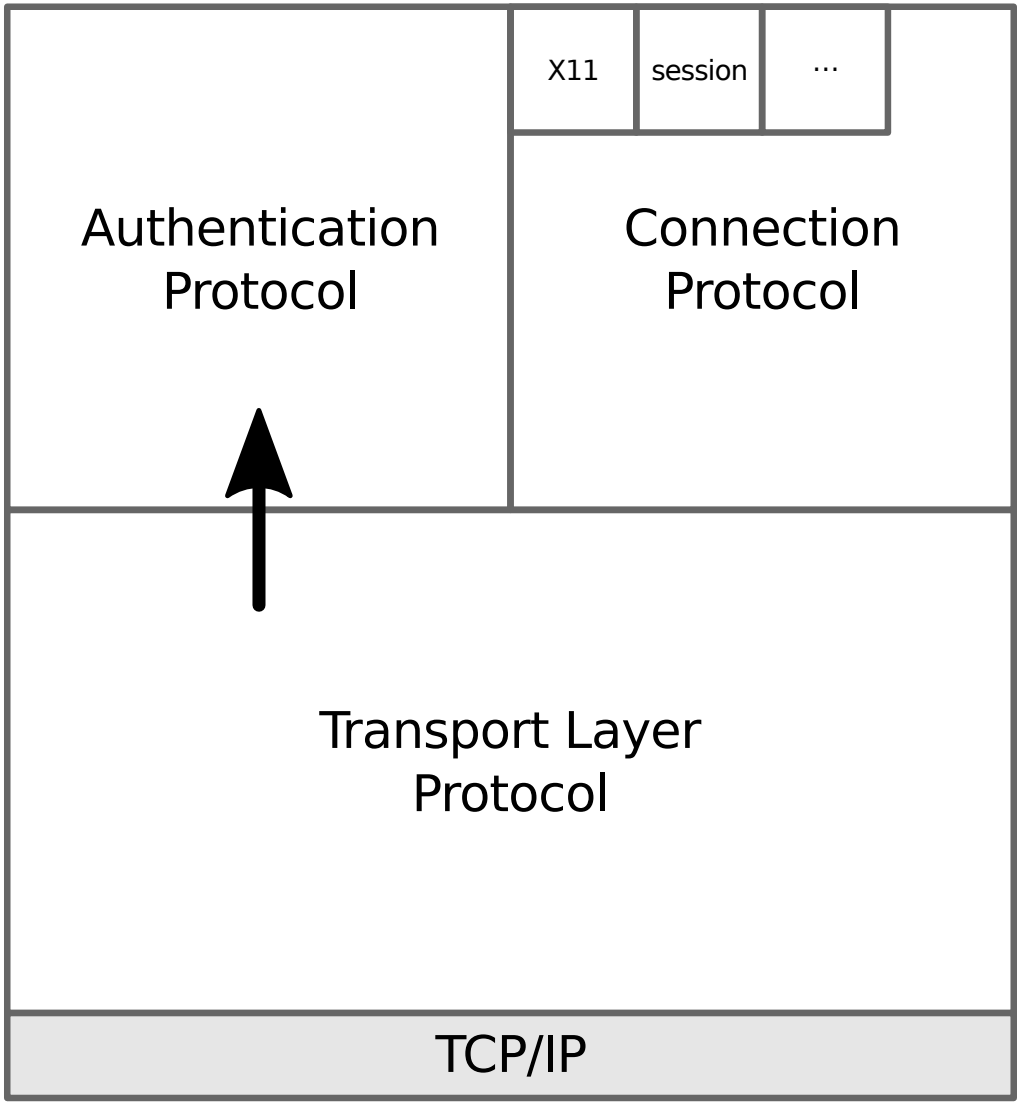
RFCs

- Kern: RFC4250–RFC4254 (5)
- Erweiterungen: RFC4255, RFC4256, RFC4335, RFC4344, RFC4345, RFC4419, RFC4462, RFC4716, RFC5656, RFC6594, RFC6668, RFC7479 (12)
- Erweiterungen (Entwurf)
- Herstellerspezifische Erweiterungen

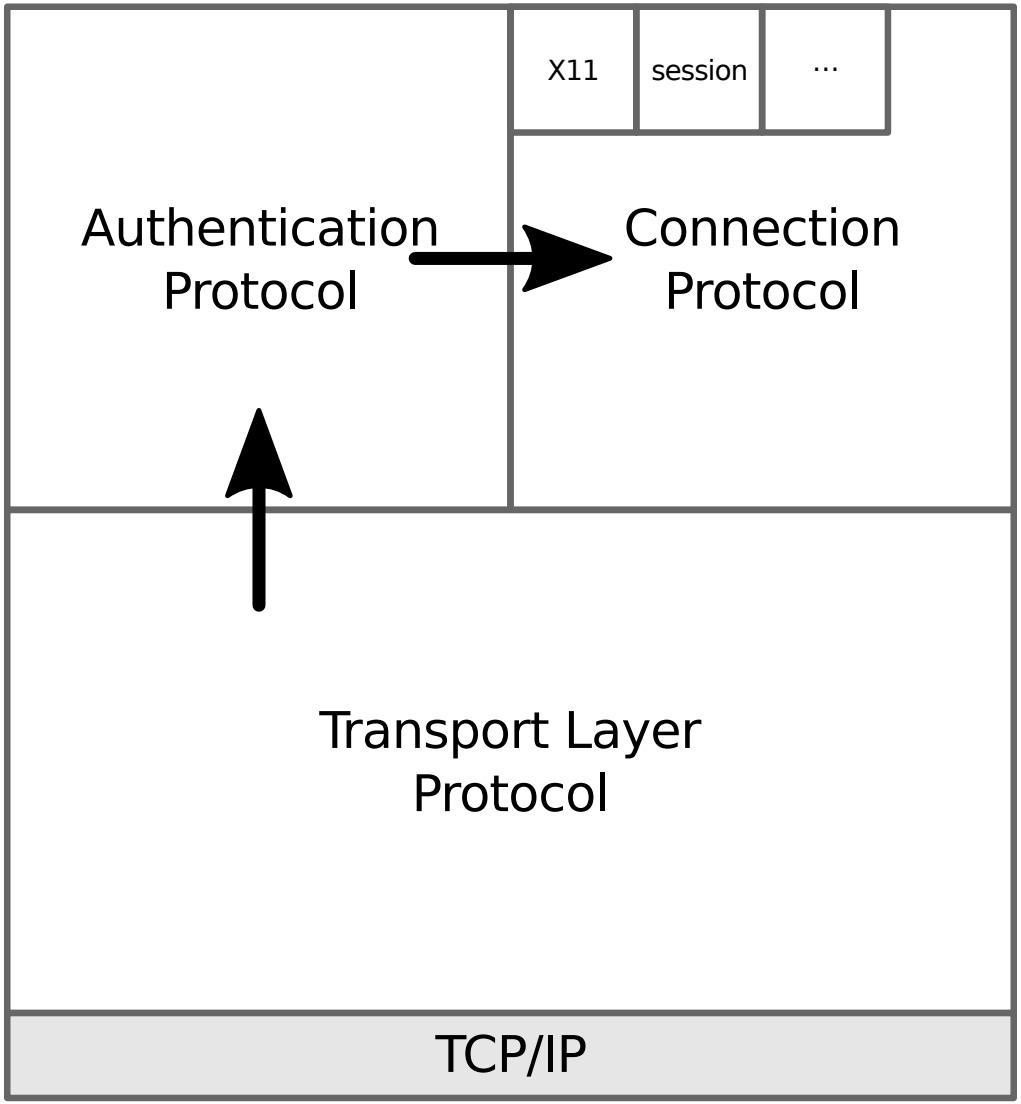
Architektur



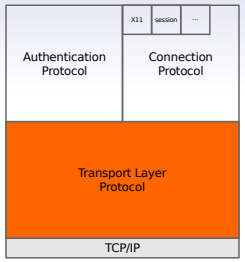
Architektur



Architektur

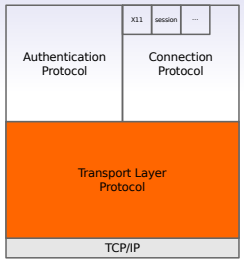


Ziele



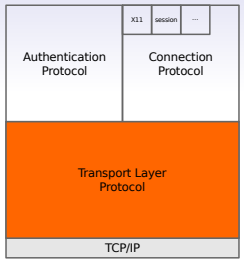
- Bereitstellung eines gesicherten Tunnels
  - Verschlüsselung
  - Integritätsschutz
  - Kompression (optional)

Ziele



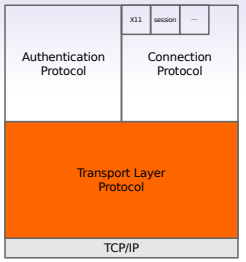
- Bereitstellung eines gesicherten Tunnels
  - Verschlüsselung
  - Integritätsschutz
  - Kompression (optional)
- Aushandlung der verwendeten Algorithmen

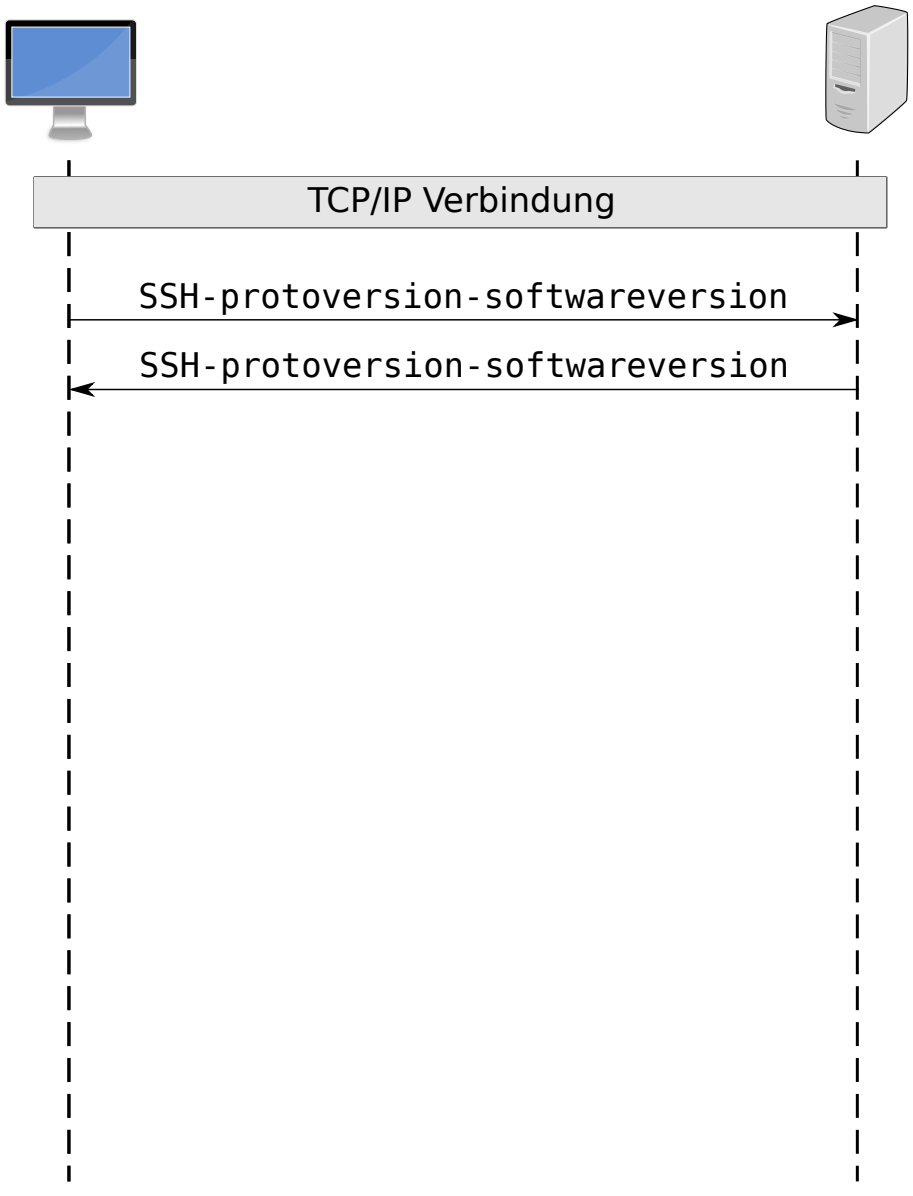
Ziele



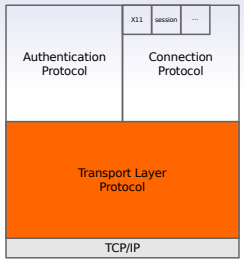
- Bereitstellung eines gesicherten Tunnels
  - Verschlüsselung
  - Integritätsschutz
  - Kompression (optional)
- Aushandlung der verwendeten Algorithmen
- Key Exchange (inkl. Server Authentifizierung)

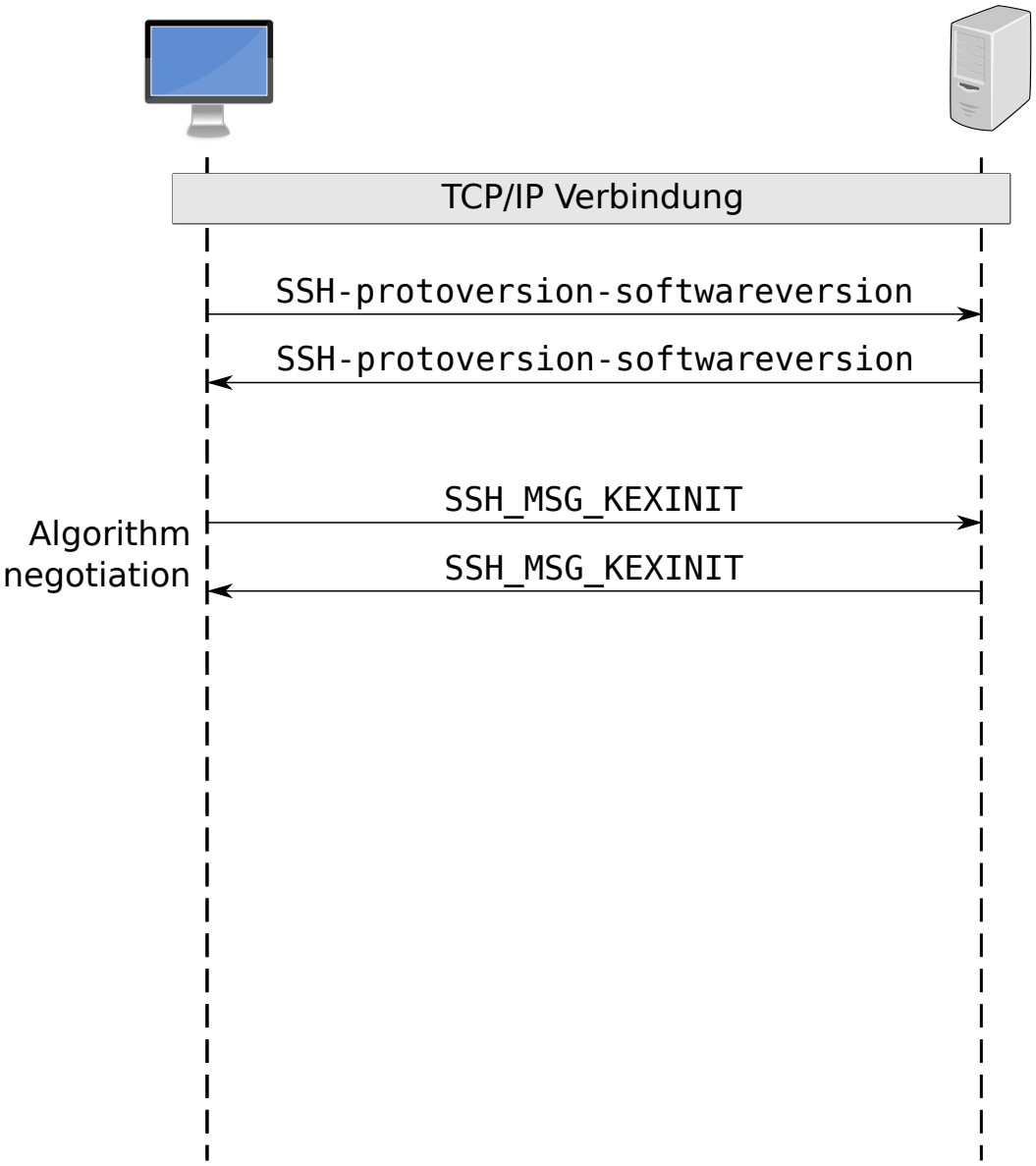
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

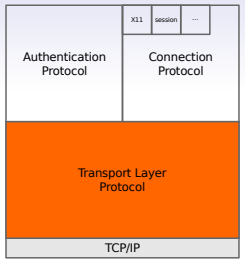

TCP/IP Verbindung
SSH-protoversion-softwareversion
SSH-protoversion-softwareversion

Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

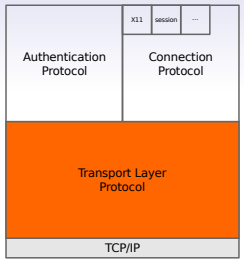

TCP/IP Verbindung
SSH-protoversion-softwareversion
SSH-protoversion-softwareversion
SSH_MSG_KEXINIT
Algorithm negotiation
SSH_MSG_KEXINIT

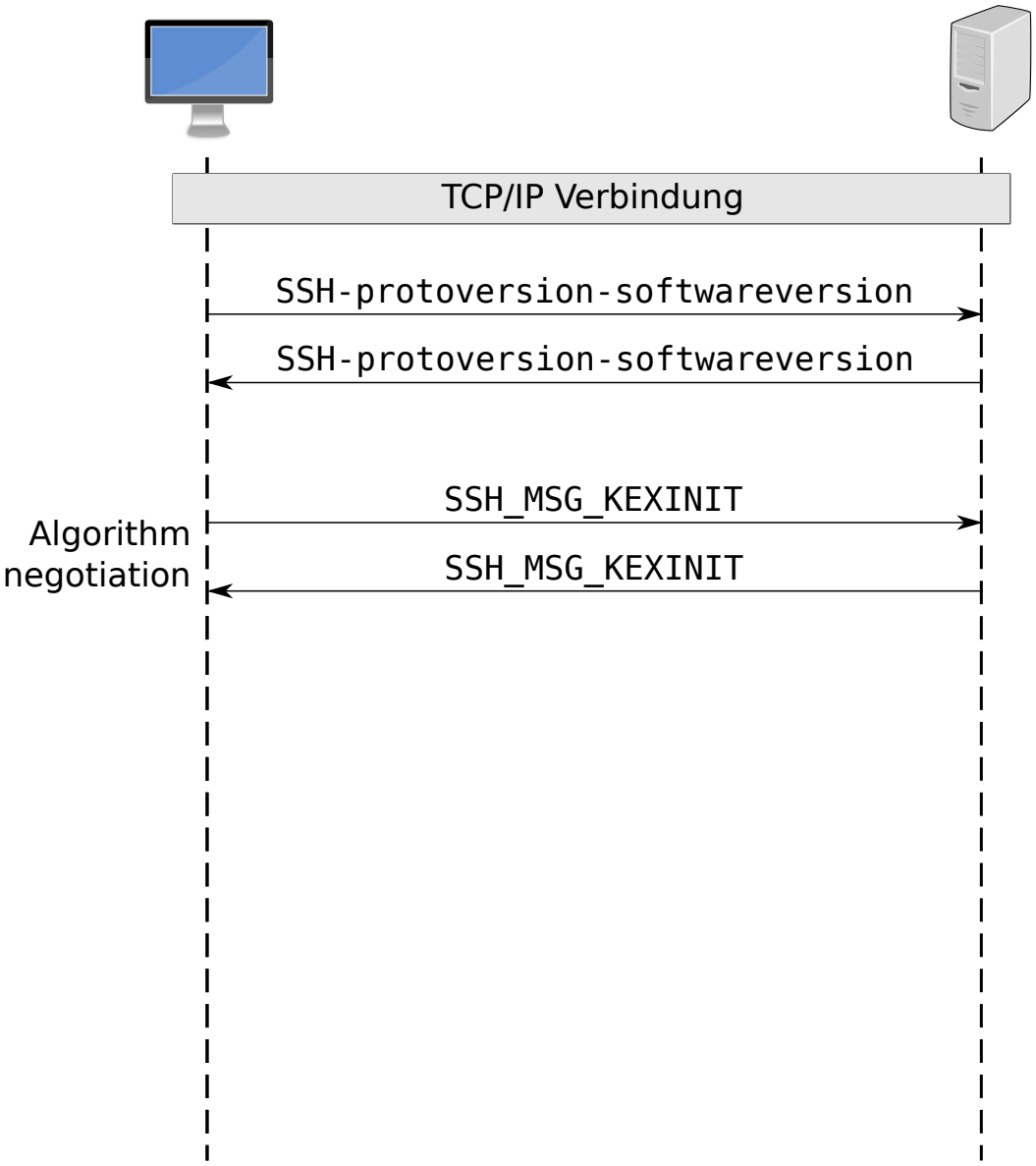
Aushandlung der Algorithmen



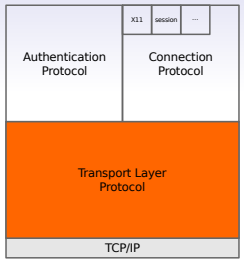
- Ausgehandelt werden u.a:
  - Key Exchange
  - Server Host Key
  - Verschlüsselung: Client $\Leftrightarrow$ Server
  - MAC: Client $\Leftrightarrow$ Server
  - Kompression: Client $\Leftrightarrow$ Server
- Die Algorithmen können sich je nach Richtung unterscheiden

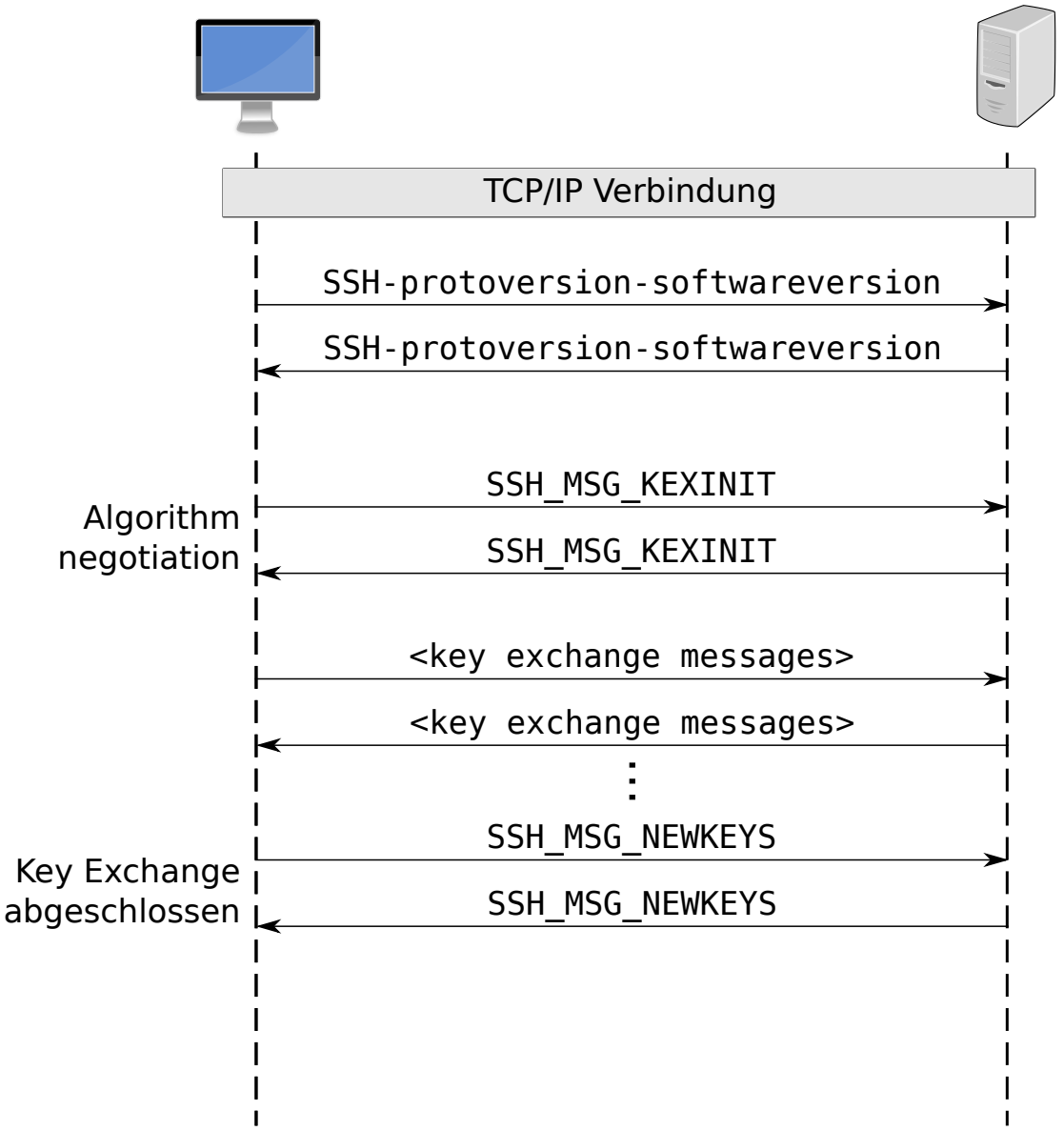
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

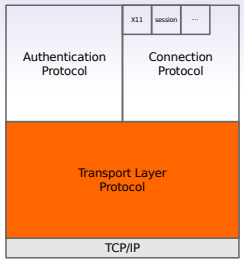

TCP/IP Verbindung
SSH-protoversion-softwareversion
SSH-protoversion-softwareversion
SSH_MSG_KEXINIT
Algorithm negotiation
SSH_MSG_KEXINIT

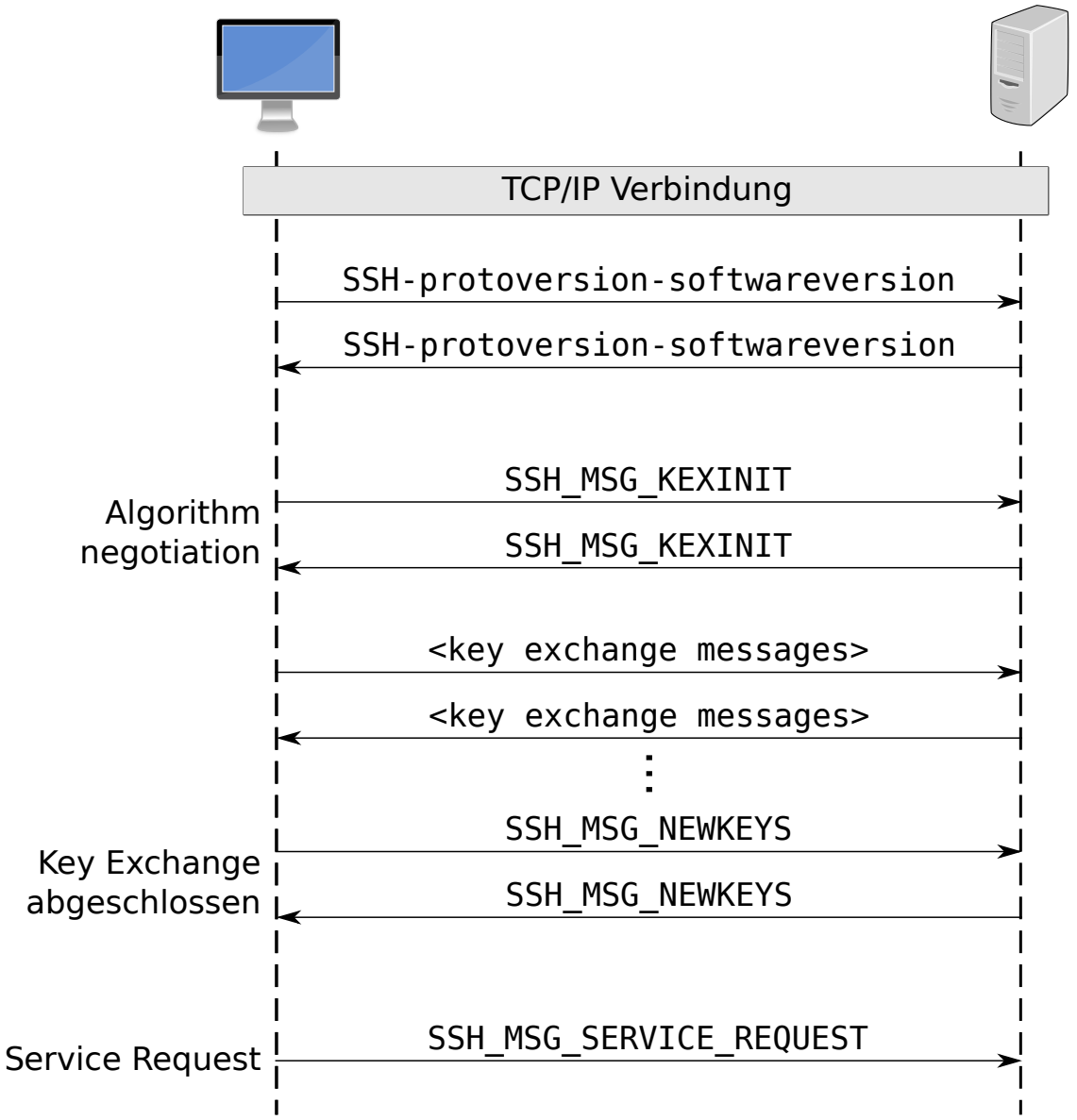
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

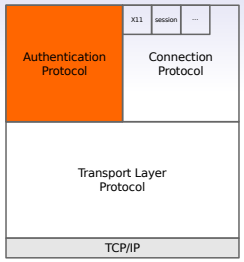

TCP/IP Verbindung
SSH-protoversion-softwareversion
SSH-protoversion-softwareversion
SSH_MSG_KEXINIT
Algorithm negotiation
SSH_MSG_KEXINIT
<key exchange messages>
<key exchange messages>
SSH_MSG_NEWKEYS
Key Exchange abgeschlossen
SSH_MSG_NEWKEYS

Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

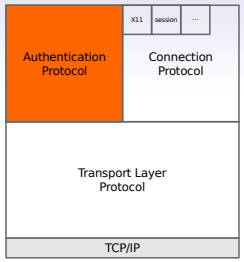

TCP/IP Verbindung
SSH-protoversion-softwareversion
SSH-protoversion-softwareversion
SSH_MSG_KEXINIT
Algorithm negotiation
SSH_MSG_KEXINIT
<key exchange messages>
<key exchange messages>
SSH_MSG_NEWKEYS
Key Exchange abgeschlossen
SSH_MSG_NEWKEYS
SSH_MSG_SERVICE_REQUEST
Service Request

Ziele



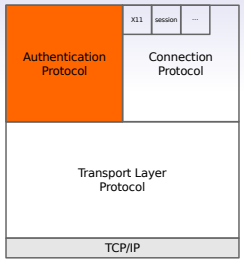
- Authentifizierung des Hosts/Benutzers gegenüber dem Server

Ziele



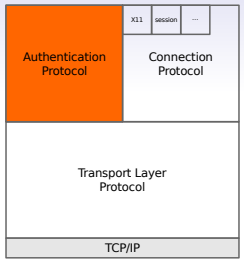
- Authentifizierung des Hosts/Benutzers gegenüber dem Server
- Die genutzten Methoden sind wählbar

Ziele



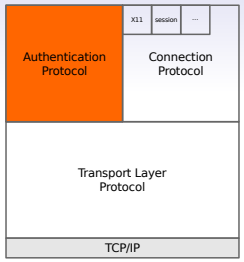
- Authentifizierung des Hosts/Benutzers gegenüber dem Server
- Die genutzten Methoden sind wählbar
- Zusätzliche Methoden sind möglich

Unterstützte Methoden



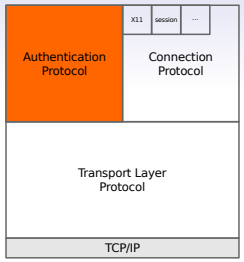
- Public key

Unterstützte Methoden



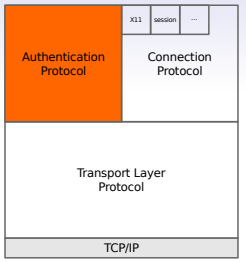
- Public key
- Password

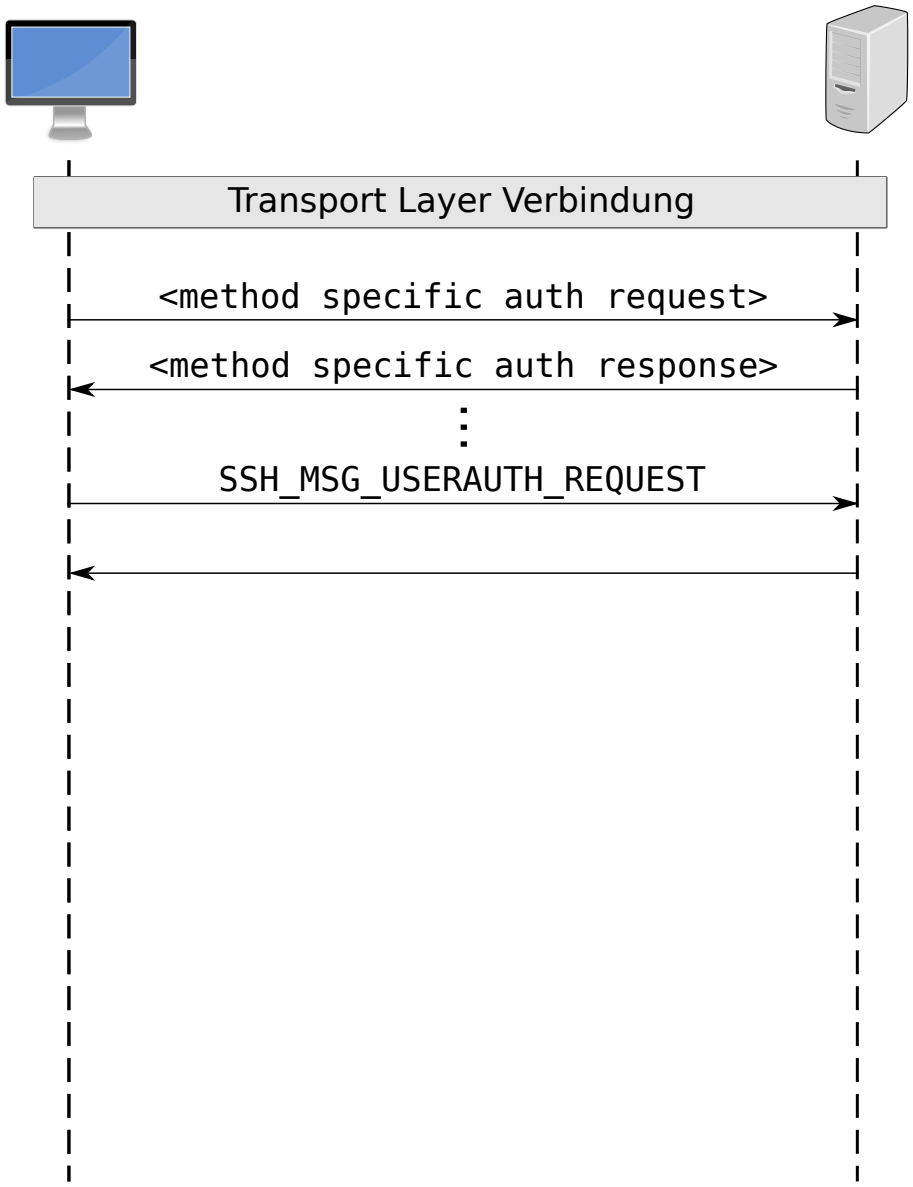
Unterstützte Methoden



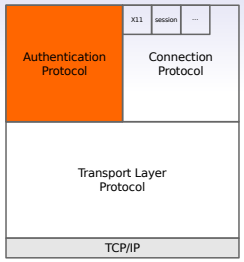
- Public key
- Password
- Host-based

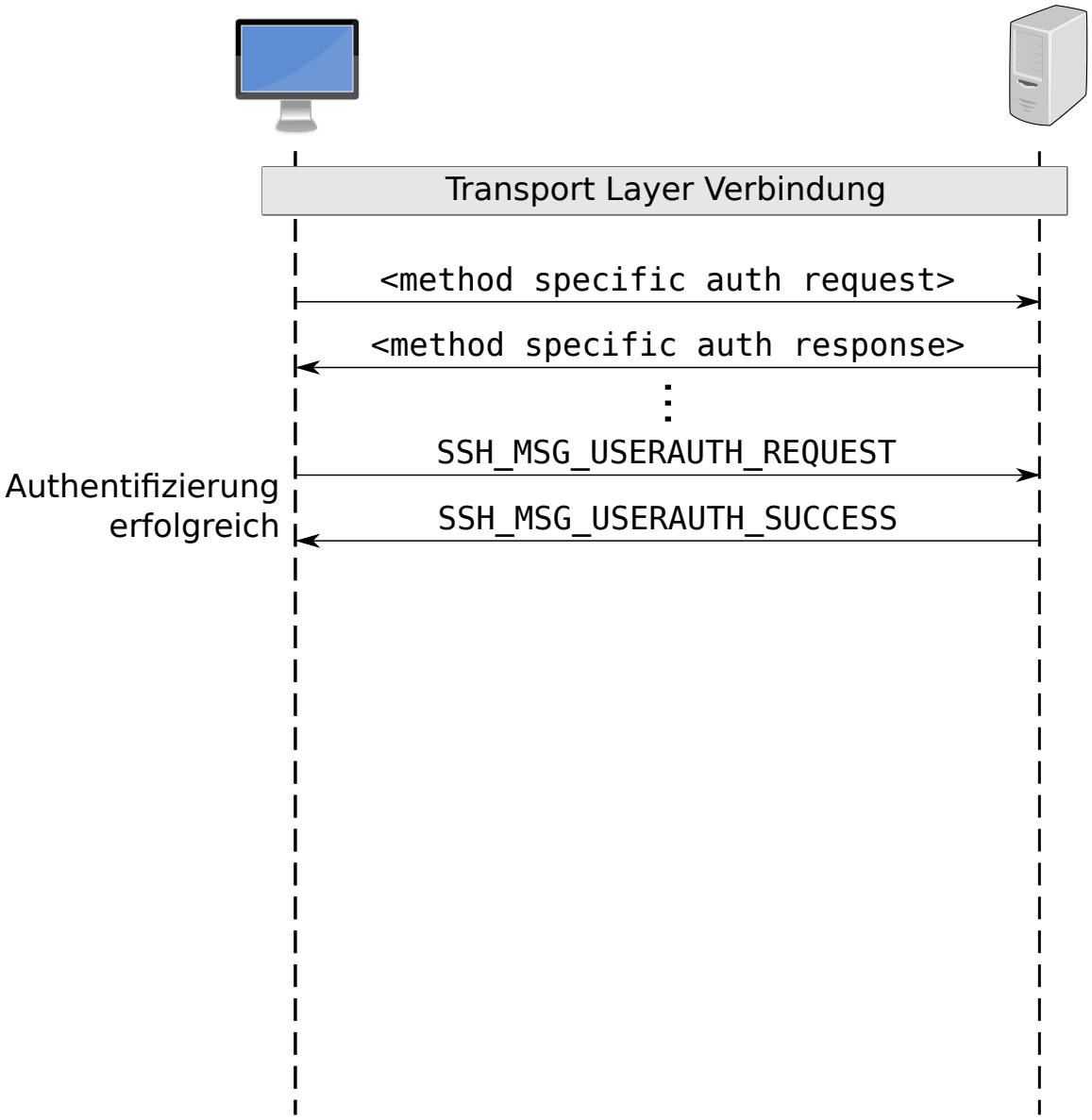
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

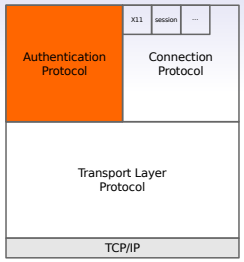

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST

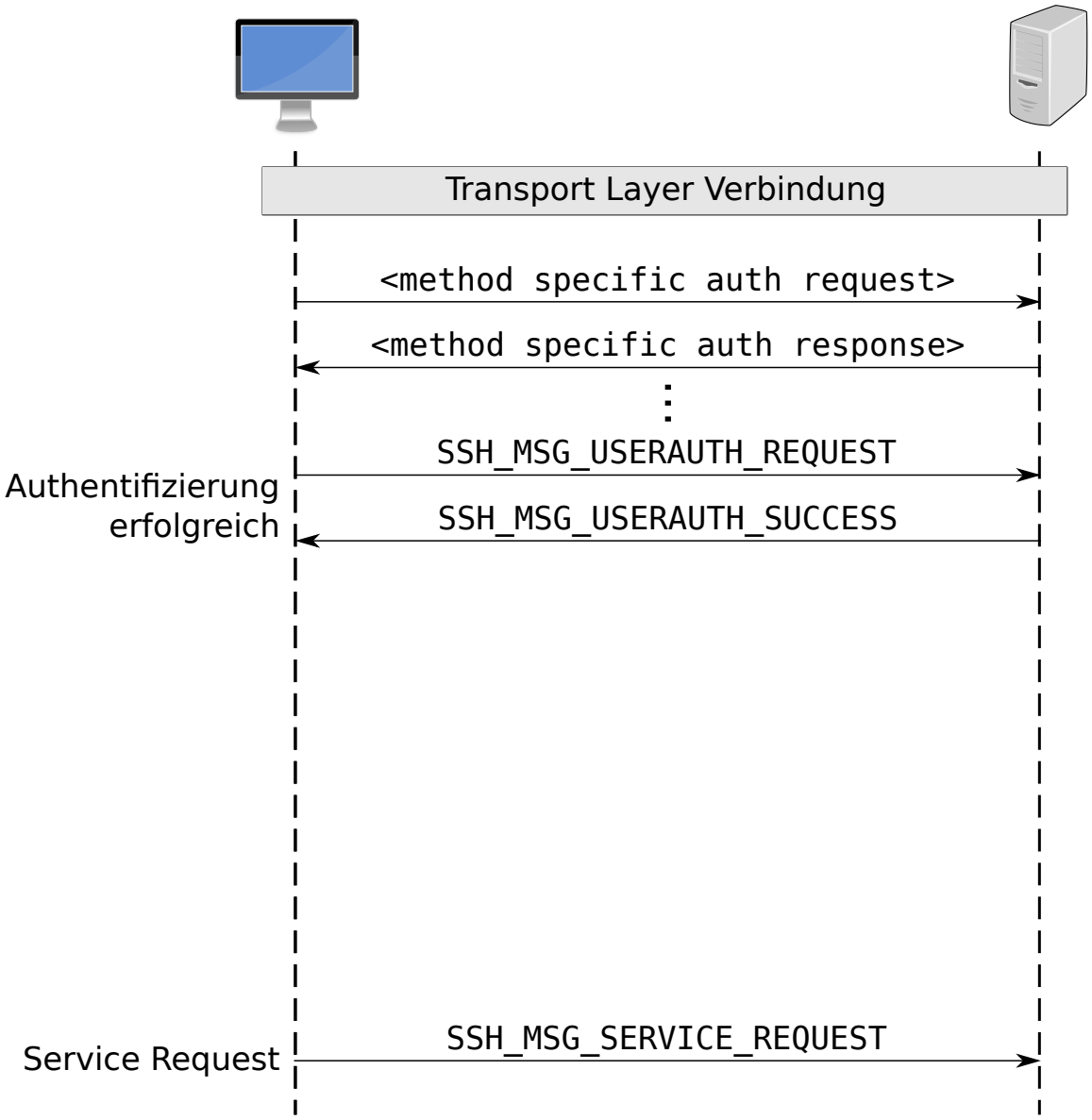
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

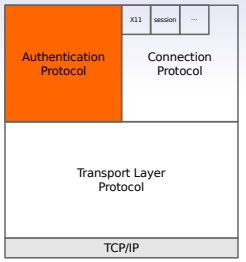

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
Authentifizierung erfolgreich
SSH_MSG_USERAUTH_SUCCESS

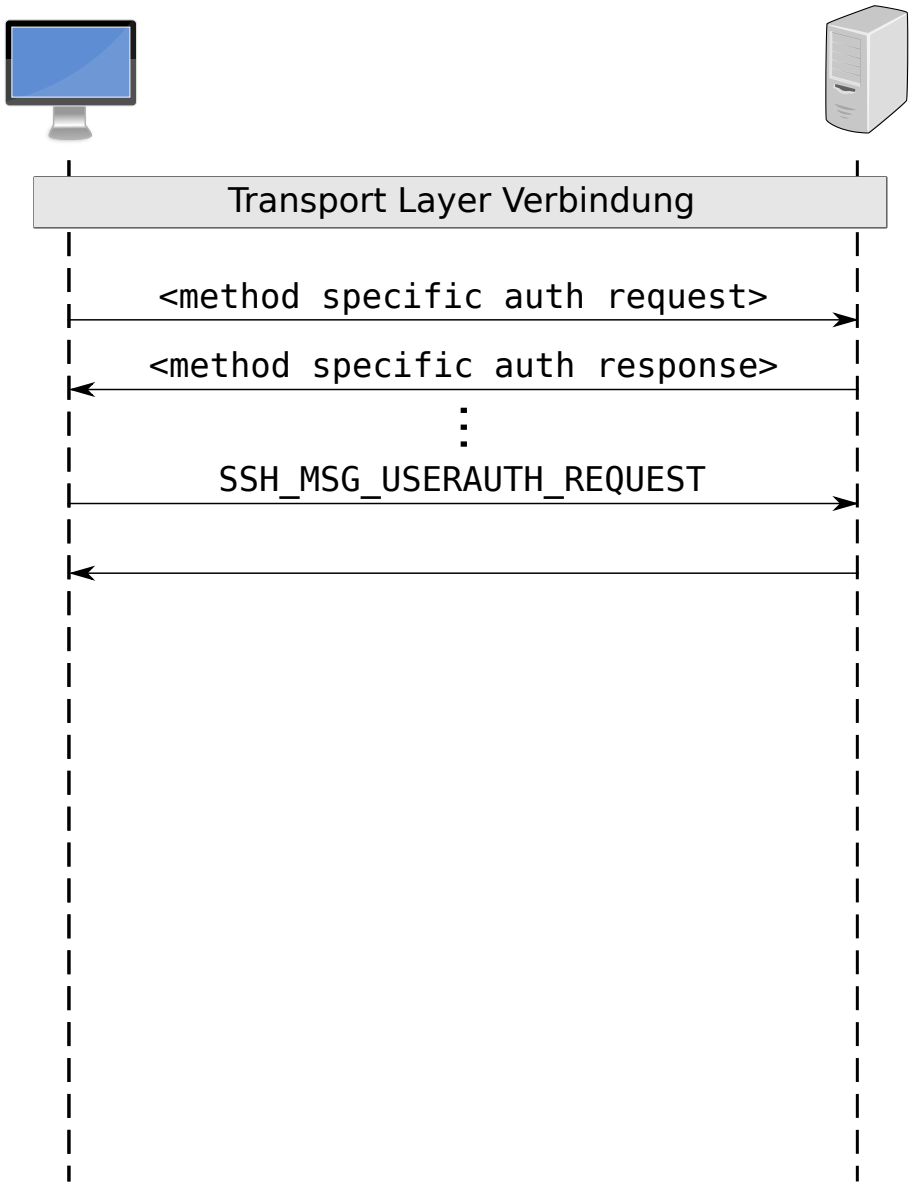
Ablauf


Authentication Protocol
X11
session
...
Connection Protocol
Transport Layer Protocol
TCP/IP

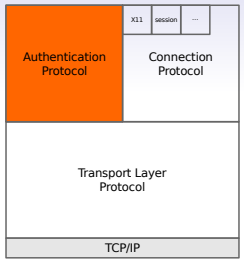

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
Authentifizierung erfolgreich
SSH_MSG_USERAUTH_SUCCESS
SSH_MSG_SERVICE_REQUEST
Service Request

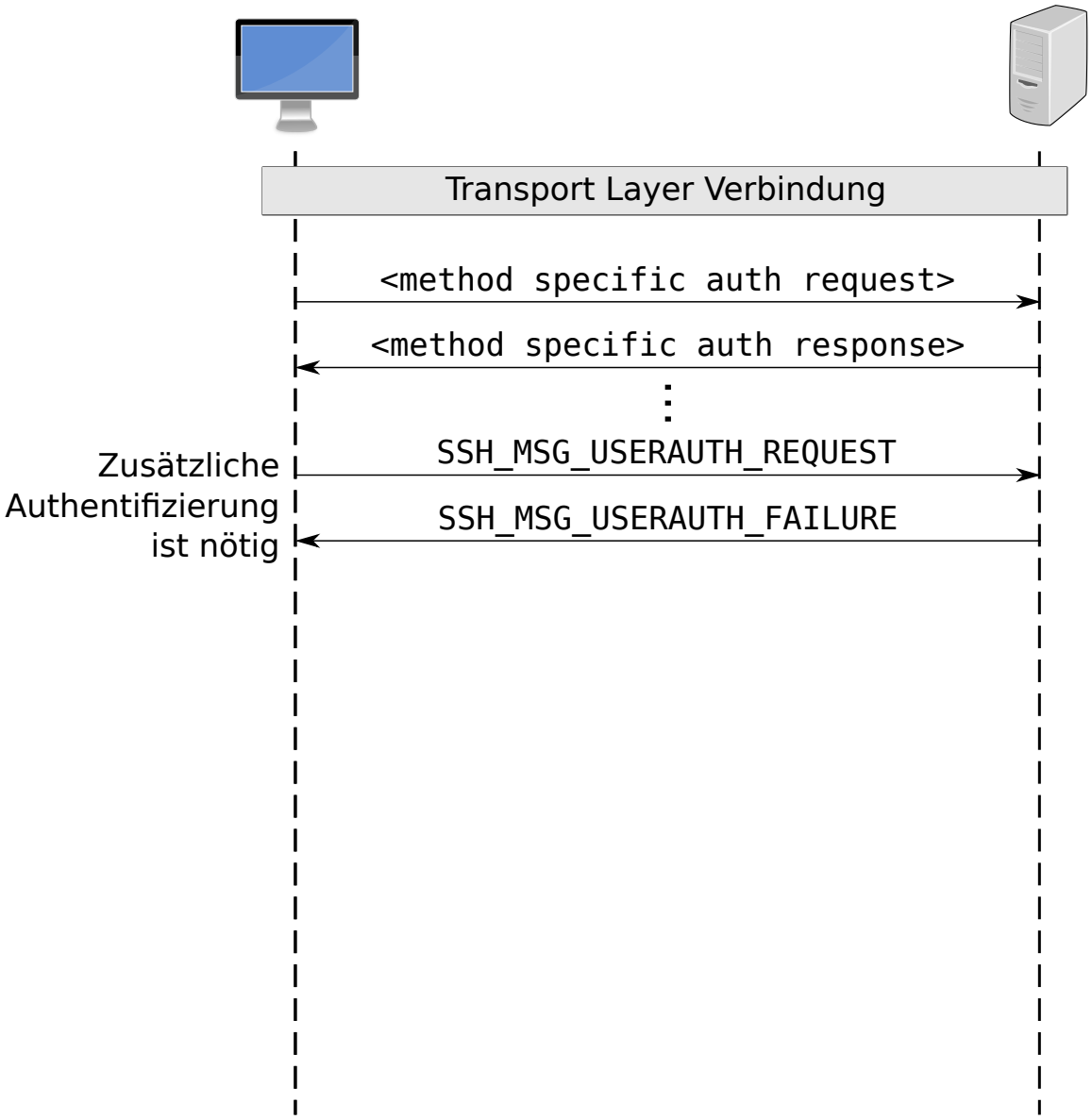
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

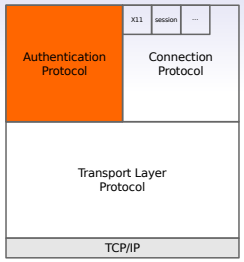

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST

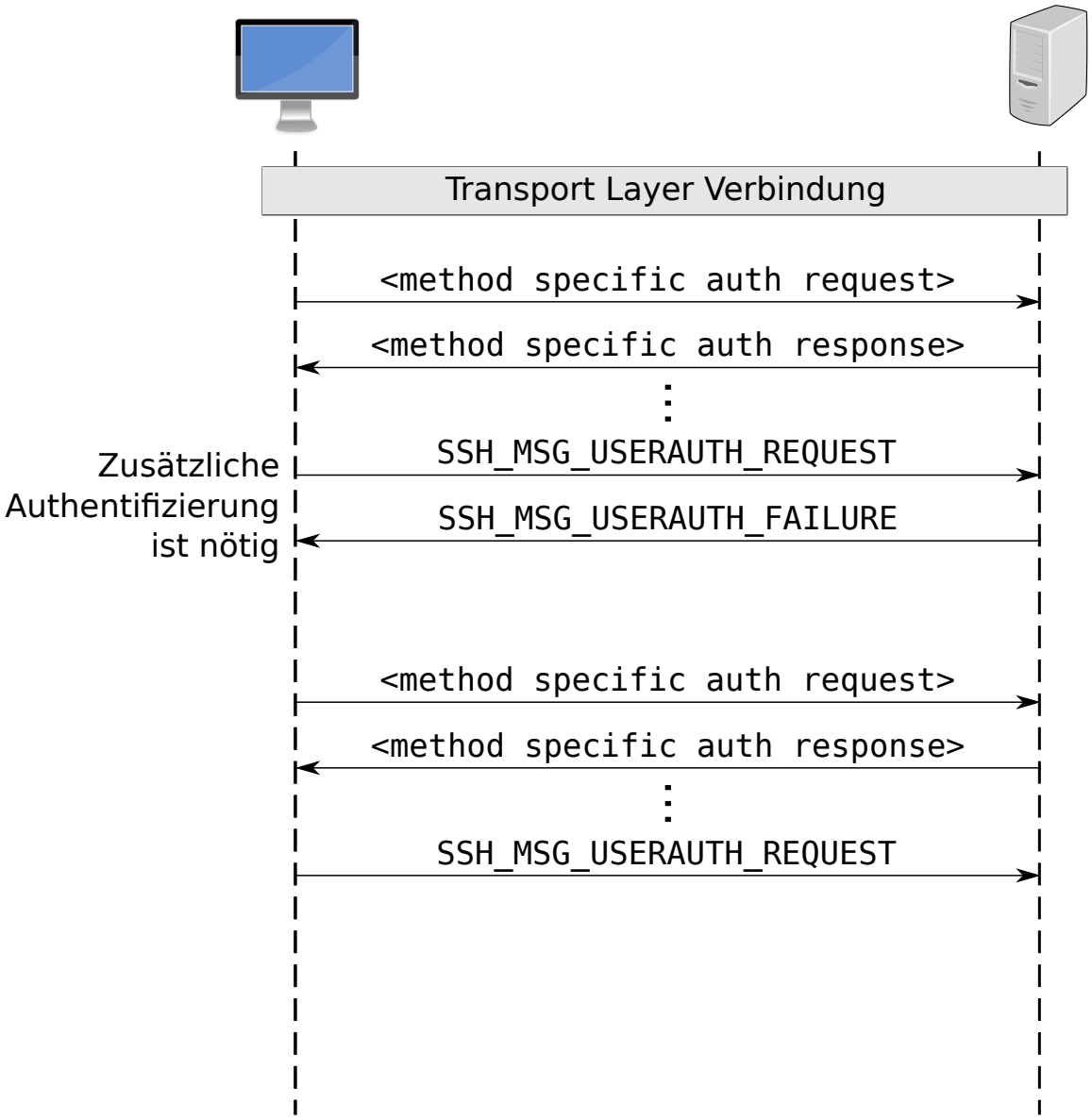
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

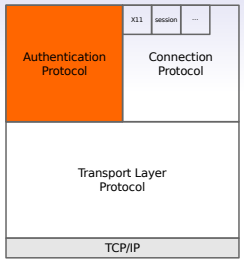

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_FAILURE
Zusätzliche Authentifizierung ist nötig

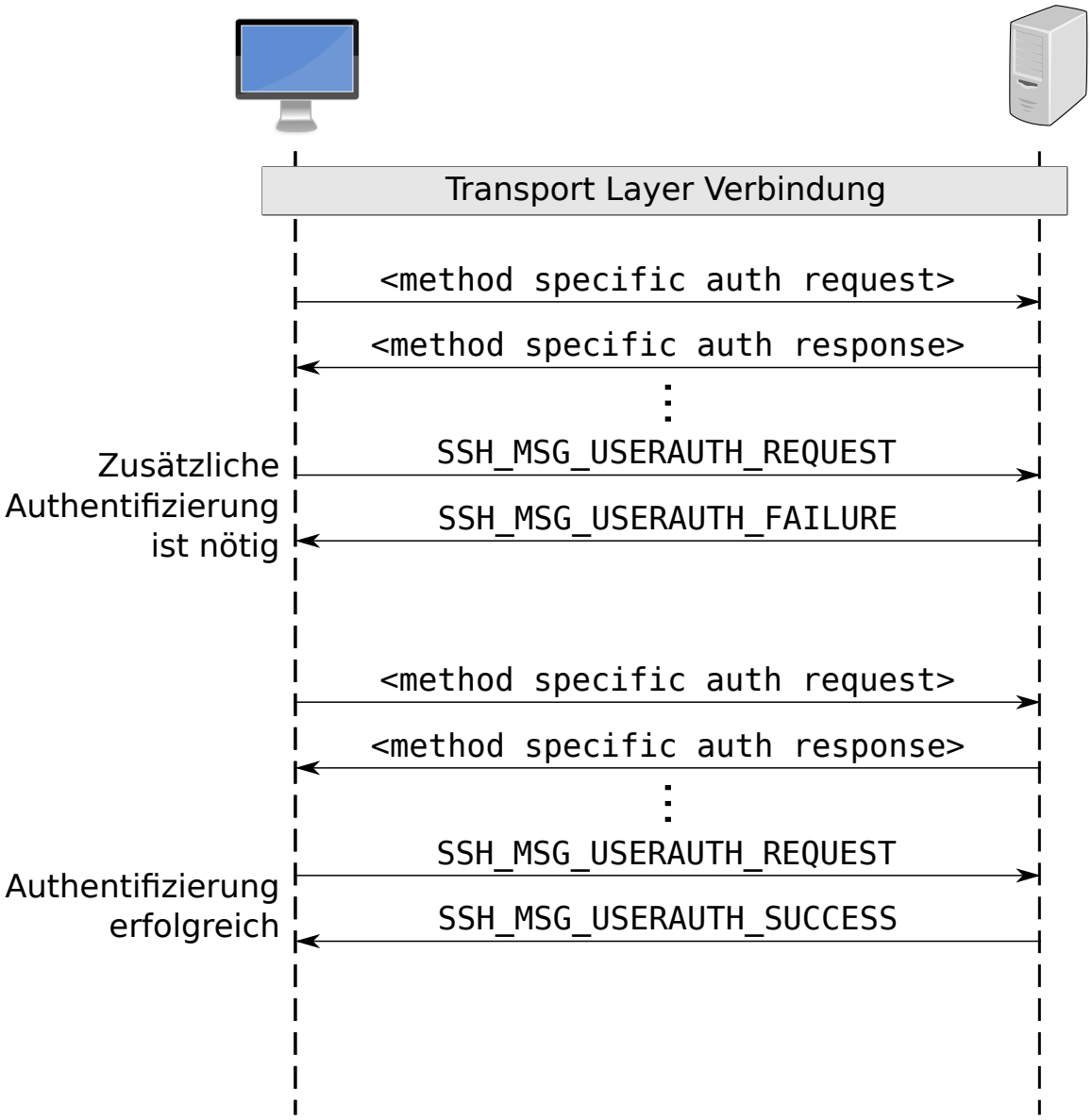
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

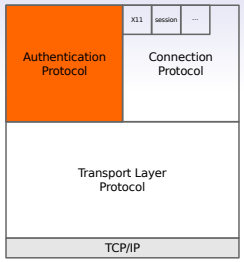

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
Zusätzliche Authentifizierung ist nötig
SSH_MSG_USERAUTH_FAILURE
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST

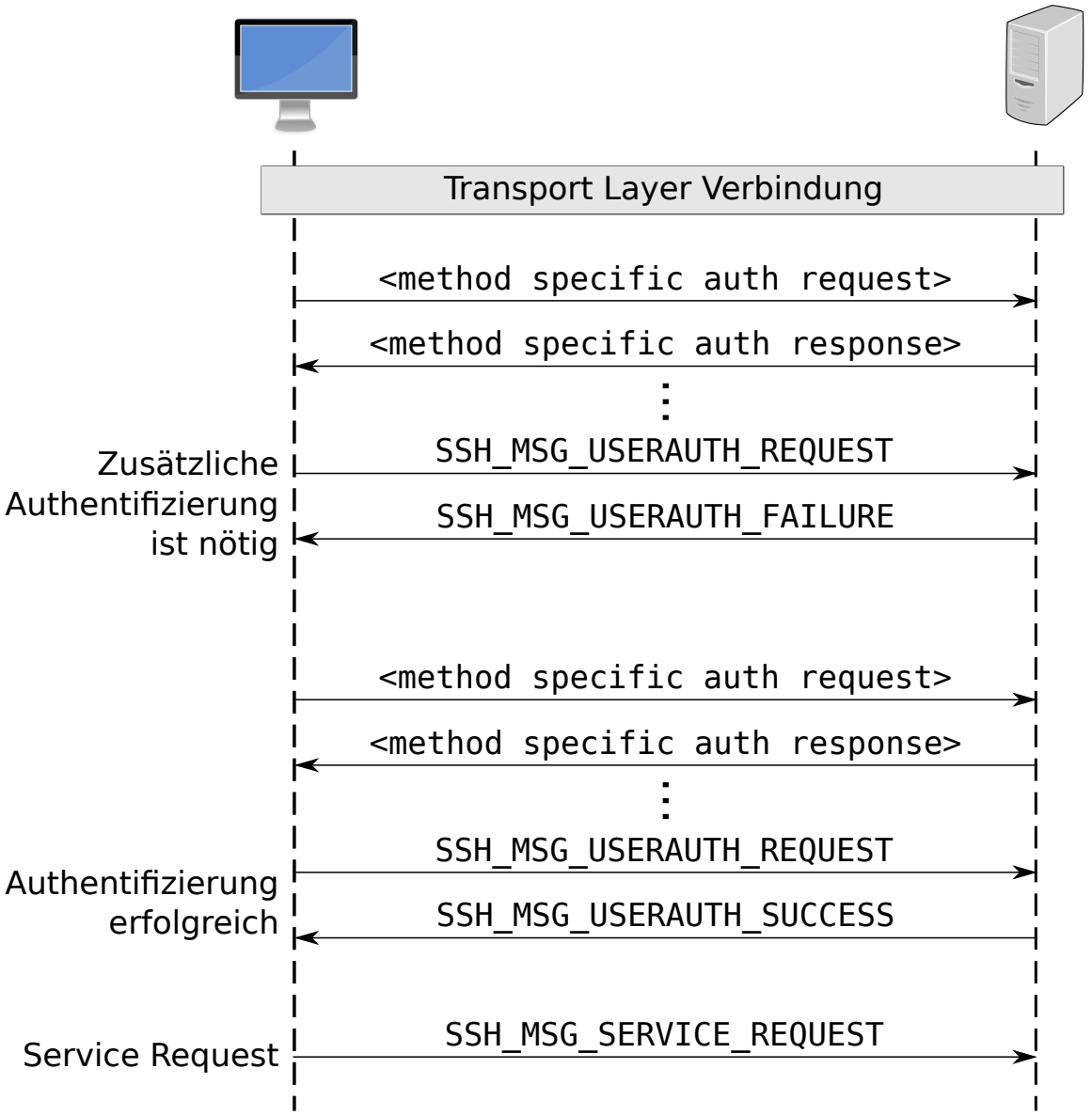
Ablauf


Authentication Protocol
X11
session
...
Connection Protocol
Transport Layer Protocol
TCP/IP

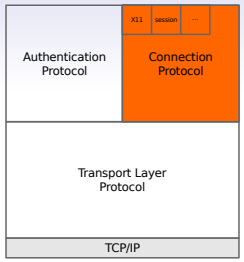

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_FAILURE
Zusätzliche Authentifizierung ist nötig
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_SUCCESS
Authentifizierung erfolgreich

Ablauf


Authentication Protocol
X11
session
...
Connection Protocol
Transport Layer Protocol
TCP/IP

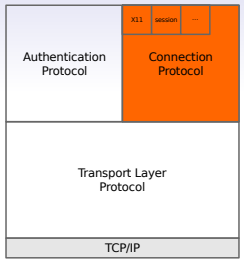

Transport Layer Verbindung
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_FAILURE
Zusätzliche Authentifizierung ist nötig
<method specific auth request>
<method specific auth response>
SSH_MSG_USERAUTH_REQUEST
SSH_MSG_USERAUTH_SUCCESS
Authentifizierung erfolgreich
SSH_MSG_SERVICE_REQUEST
Service Request

Ziele



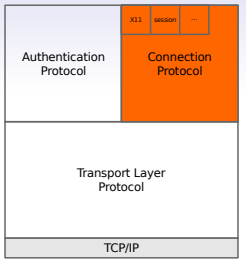
- Bereitstellung von Kanälen:
  - Interaktive Sessions
  - Ausführen von Kommandos
  - X11 Forwarding
  - TCP/IP Forwarding

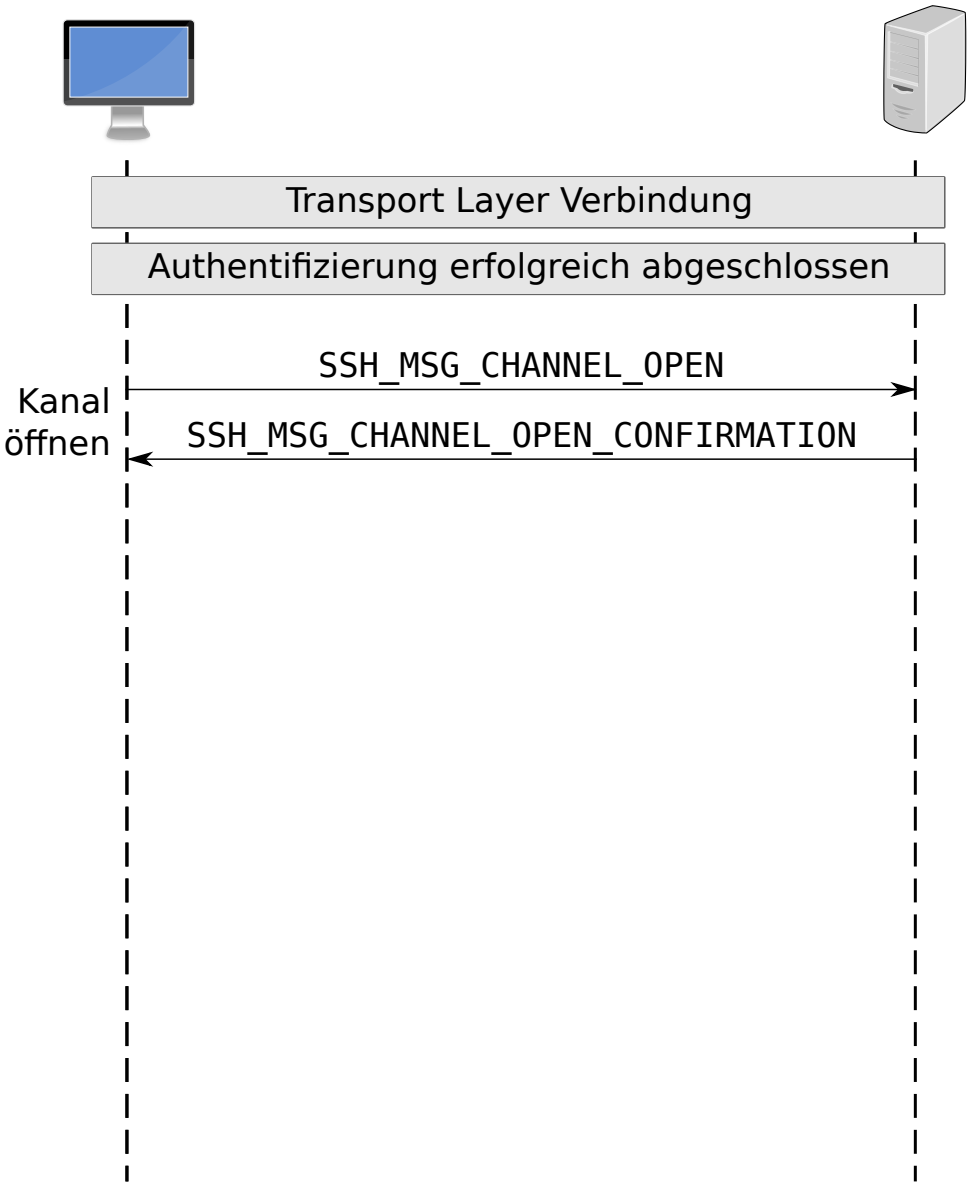
Ziele



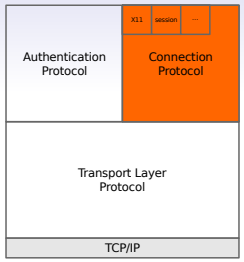
- Bereitstellung von Kanälen:
  - Interaktive Sessions
  - Ausführen von Kommandos
  - X11 Forwarding
  - TCP/IP Forwarding
- Kanäle werden über den sicheren Tunnel multiplexed

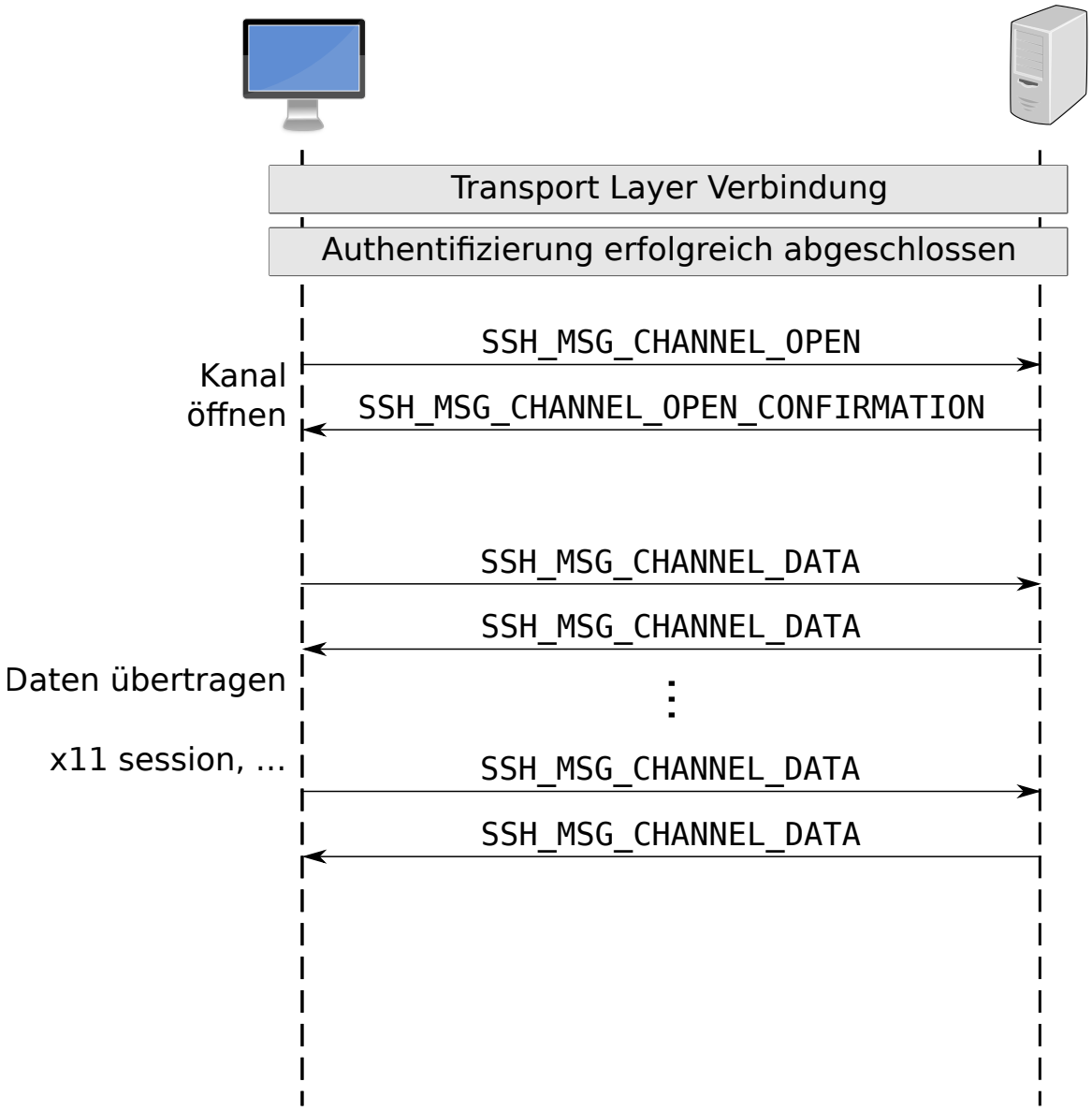
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP

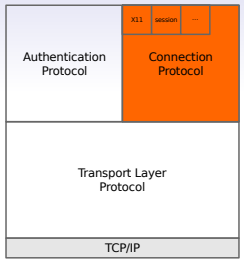

Transport Layer Verbindung
Authentifizierung erfolgreich abgeschlossen
SSH_MSG_CHANNEL_OPEN
Kanal öffnen
SSH_MSG_CHANNEL_OPEN_CONFIRMATION

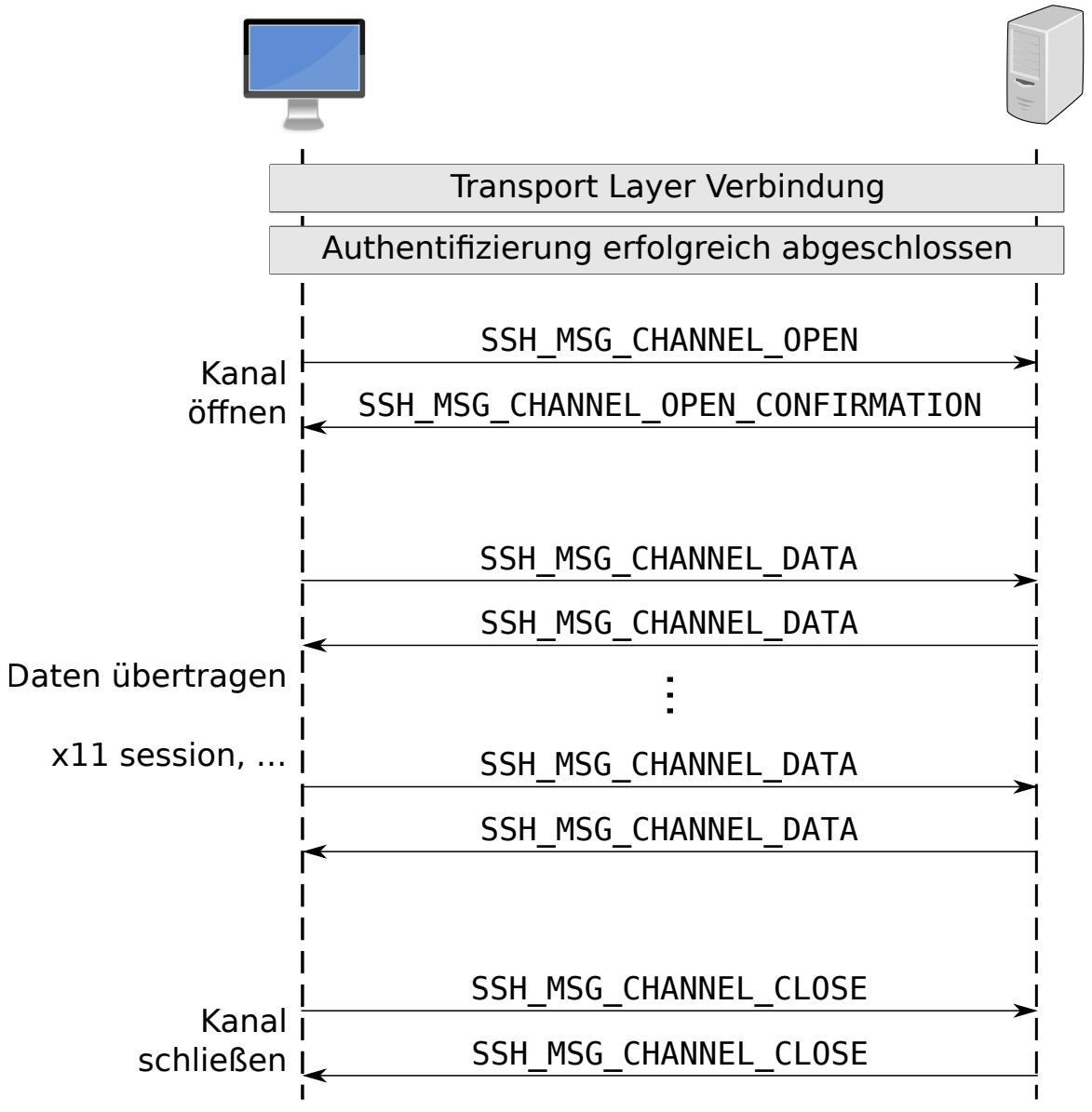
Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP


Transport Layer Verbindung
Authentifizierung erfolgreich abgeschlossen
SSH_MSG_CHANNEL_OPEN
Kanal öffnen
SSH_MSG_CHANNEL_OPEN_CONFIRMATION
SSH_MSG_CHANNEL_DATA
SSH_MSG_CHANNEL_DATA
Daten übertragen
x11 session, ...
SSH_MSG_CHANNEL_DATA
SSH_MSG_CHANNEL_DATA

Ablauf


X11
session
...
Authentication Protocol
Connection Protocol
Transport Layer Protocol
TCP/IP


Transport Layer Verbindung
Authentifizierung erfolgreich abgeschlossen
SSH_MSG_CHANNEL_OPEN
Kanal öffnen
SSH_MSG_CHANNEL_OPEN_CONFIRMATION
SSH_MSG_CHANNEL_DATA
SSH_MSG_CHANNEL_DATA
Daten übertragen
x11 session, ...
SSH_MSG_CHANNEL_DATA
SSH_MSG_CHANNEL_DATA
SSH_MSG_CHANNEL_CLOSE
Kanal schließen
SSH_MSG_CHANNEL_CLOSE

Demo

Fazit

- Bereitstellung von sicheren Diensten über ein ungesichertes Netzwerk

Fazit

- Bereitstellung von sicheren Diensten über ein ungesichertes Netzwerk
- Drei Schichten:
  - Transport Layer Protocol
  - Authentication Protocol
  - Connection Protocol

Fazit

- Bereitstellung von sicheren Diensten über ein ungesichertes Netzwerk
- Drei Schichten:
  - Transport Layer Protocol
  - Authentication Protocol
  - Connection Protocol
- Erweiterbar & flexibel

Fazit

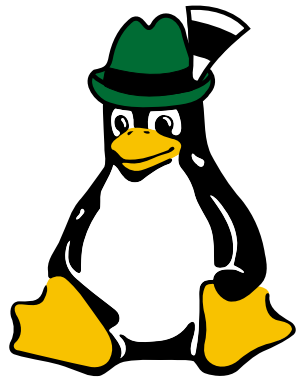
- Bereitstellung von sicheren Diensten über ein ungesichertes Netzwerk
- Drei Schichten:
  - Transport Layer Protocol
  - Authentication Protocol
  - Connection Protocol
- Erweiterbar & flexibel
- Ausprobieren: `ssh -v[vv] user@host`

Ein Blick auf SSH

Florian Preinstorfer

https://nblock.org

Grazer Linuxtage
29.04.2017



Quellen

- RFC4250–RFC4254
- OpenSSH Webseite `https://www.openssh.com`
- Protocol Basics: Secure Shell Protocol `https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-46/124-ssh.html`
- OK Icon: `https://openclipart.org/detail/212394/ok` (sixsixfive, Public Domain)
- Cancel Icon: `https://openclipart.org/detail/211689/matticonscancel` (sixsixfive, Public Domain)
- Modern Desktop: `https://openclipart.org/detail/166823/modern-desktop` (gsagri04, Public Domain)

Quellen

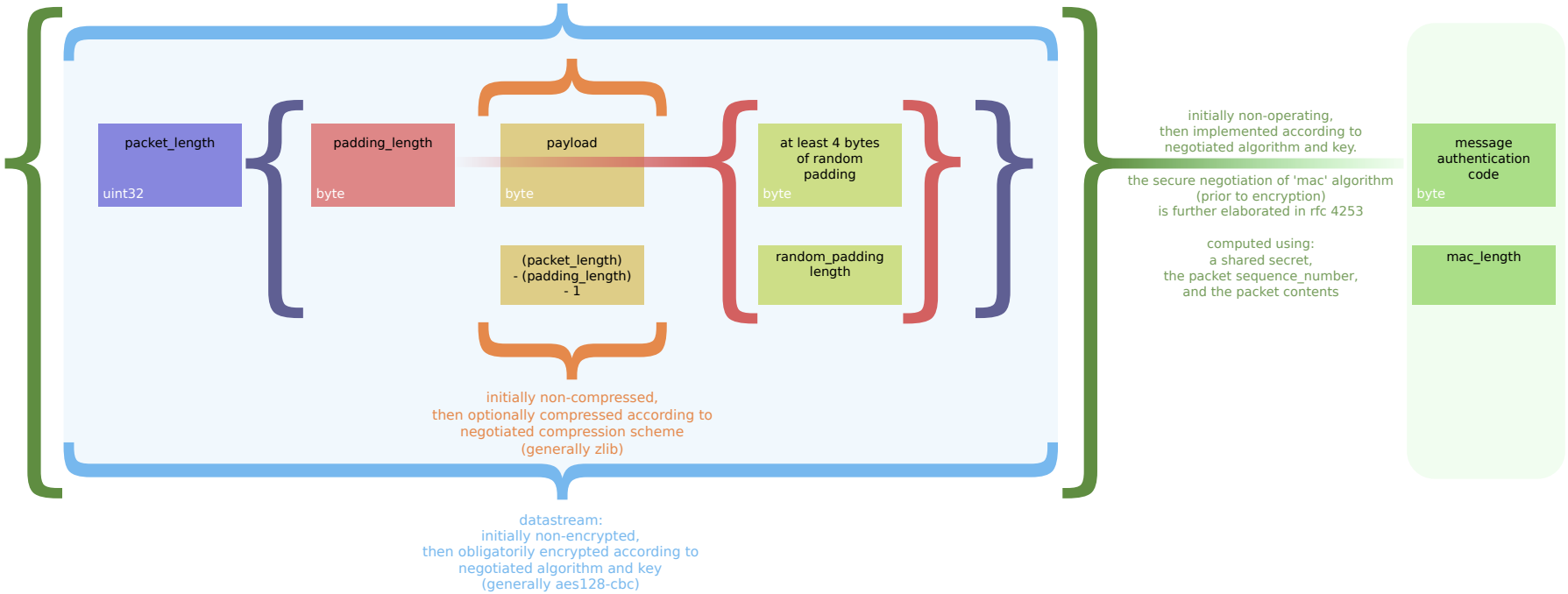
- Server: `https://openclipart.org/detail/155101/server` (saisyukusanagi, Public Domain)
- SSH Paket: `https://en.wikipedia.org/wiki/File:Ssh_binary_packet_alt.svg` (User:Stannered, CC-BY-SA 2.5)
- DH Key Exchange: `https://upload.wikimedia.org/wikipedia/commons/4/46/Diffie-Hellman_Key_Exchange.svg` (A.J. Han Vinck, Public Domain)

```
flo@notizblock-laptop ~ $ ssh florian@nblock.org
The authenticity of host 'nblock.org (37.120.166.62)' can't be established.
ECDSA key fingerprint is SHA256:d2FpCOGQfUTEA1tW8vKf8zqp8G45vmiHgUr4xBP0jck.
Are you sure you want to continue connecting (yes/no)?
```

Erweiterbarkeit

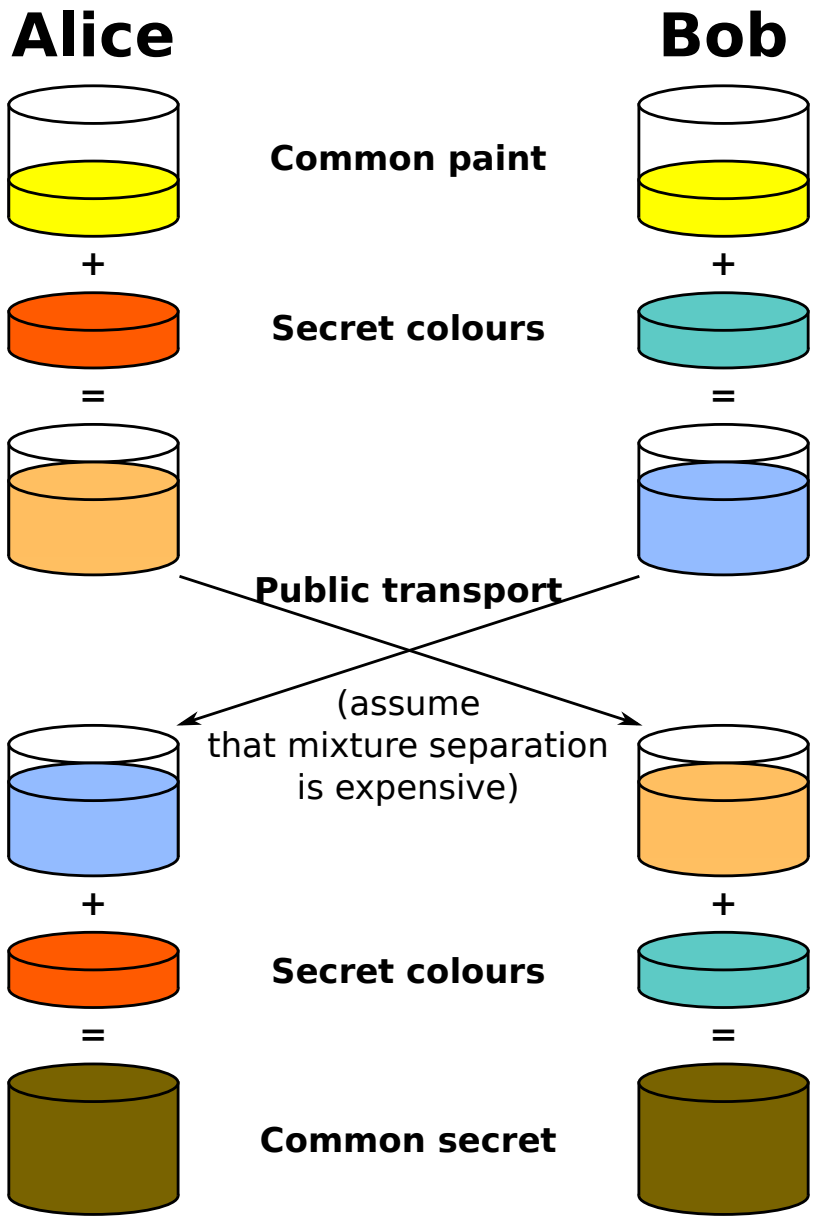
- Im Standard spezifiziert
  - 3des-cbc
  - blowfish-cbc
  - ...
- Herstellerspezifisch
  - ourcipher-cbc@example.com
  - chacha20-poly1305@openssh.com
  - ...

ssh2 binary packet 35000 bytes or less



sequence_number

...is an implicit packet sequence number represented as uint32. The packet sequence_number itself is not included in the packet sent over the wire. The sequence_number is initialized to zero for the first packet, and is incremented after every packet (regargless of whether encryption or MAC is in use). It is never reset (even if keys/algorithms are renegotiated later!). It wraps around to zero after every 2^32 packets.


Alice
Bob
Common paint
+
Secret colours
=
Public transport
(assume
that mixture separation
is expensive)
+
Secret colours
=
Common secret